



Was ist Paperclip? Voraussetzungen 1 · Installation 2 · Claude-Login 3 · Erster Agent
4 · Codex-Agent 5 · Erste Aufgabe Bekannte Probleme Effizienz-Tipps


Paperclip einrichten – ganz ohne Vorkenntnisse

Diese Anleitung führt dich Schritt für Schritt durch die Einrichtung deiner eigenen KI-Agenten-Plattform auf einem eigenen Server.

 Claude Max · Codex

 Hetzner · Coolify

 ca. 30–60 Min

 Kein Coding-Wissen nötig

EINFÜHRUNG

Was ist Paperclip?

Stell dir Paperclip vor wie ein **Projektmanagement-Tool** – ähnlich wie Trello oder ClickUp – aber mit einem entscheidenden Unterschied: Die Aufgaben werden **automatisch von KI-Agenten erledigt**, nicht von Menschen.

Du schreibst eine Aufgabe, zum Beispiel: *"Schreibe einen Instagram-Post über KI"*. Dein Agent übernimmt, denkt nach, schreibt den Post, und meldet dir das Ergebnis zurück.

Die wichtigsten Begriffe

Agent

Ein KI-Mitarbeiter mit einer festen Rolle – z.B. "Texter",

Issue

Eine einzelne Aufgabe, die dem Agenten zugewiesen wird

Run

Ein Durchlauf: der Agent liest das Issue und bearbeitet es

"Forscher", "CMO"

Adapter

Das KI-Modell hinter dem Agenten: Claude oder Codex

AGENTS.md

Die "Jobbeschreibung" des Agenten – kurz halten!

Heartbeat

Automatischer Rhythmus: der Agent prüft regelmäßig ob neue Aufgaben da sind



Screenshot: Paperclip Dashboard – Gesamtansicht

Agents-Übersicht mit CMO, Tester und Codex

Ersetze diesen Bereich durch einen Screenshot der Paperclip-Oberfläche

VORBEREITUNG

Was du brauchst

- Server** – empfohlen: Hetzner VPS (CX22, ~5€/Monat) mit Ubuntu 22.04. Kein eigener Computer nötig, alles läuft in der Cloud.
- Coolify** – kostenlose Software auf deinem Server, die Docker-Container verwaltet. Installation dauert 5 Minuten (coolify.io).
- Claude Max Abo** (~\$100/Monat) für Claude-Agenten. Kein API-Key nötig – du loggst dich mit deinem claude.ai-Account ein.
- ChatGPT Pro/Plus Abo** (optional, ~\$20–200/Monat) falls du auch den Codex-Agenten nutzen möchtest.
- DNS-Eintrag** (optional, empfohlen) – eine eigene Domain, die auf deinen Server zeigt, z.B. `paperclip.deine-domain.de`.



KOSTEN GESAMT

Server (~5€) + Claude Max (~100\$) = ca. 105€/Monat für einen voll funktionsfähigen KI-Agenten, der rund um die Uhr arbeitet.

SCHRITT 1

1 Paperclip installieren

Die Installation läuft über **Docker** – ein System, das Software in isolierten "Containern" ausführt. Du musst Docker nicht verstehen, du gibst nur die folgenden Befehle ein.

1 Coolify-Terminal öffnen

In Coolify (linkes Menü) auf **Terminal** klicken, Server "localhost" auswählen, auf **Connect** klicken.



Screenshot: Coolify Terminal – Verbindungsbildschirm

Terminal-Seite mit "localhost" ausgewählt und Connect-Button

2 Datenbank starten

Diesen Befehl ins Terminal kopieren und Enter drücken:

```
TERMINAL – DATENBANK

docker run -d \
  --name paperclip-db \
  --network coolify \
  -e POSTGRES_USER=paperclip \
  -e POSTGRES_PASSWORD=paperclip \
  -e POSTGRES_DB=paperclip \
  postgres:16-alpine
```

3

Paperclip starten

Diesen Befehl eingeben. **Ersetze dabei** DEIN-GEHEIMER-SCHLÜSSEL durch eine beliebige Zeichenkette (z.B. mein-super-geheimes-passwort-2025) und DEINE-IP-ODER-DOMAIN durch deine Server-IP oder Domain:

TERMINAL – PAPERCLIP

```
docker run -d \  
  --name paperclip \  
  --network coolify \  
  --security-opt seccomp=unconfined \  
  -p 3100:3100 \  
  -e DATABASE_URL=postgresql://paperclip:paperclip@paperclip-  
db:5432/paperclip \  
  -e PORT=3100 \  
  -e BETTER_AUTH_SECRET=DEIN-GEHEIMER-SCHLÜSSEL \  
  -e BETTER_AUTH_URL=http://DEINE-IP-ODER-DOMAIN:3100 \  
  ghcr.io/paperclipai/paperclip:latest
```



WICHTIG

Füge NIEMALS `--user root` in diesen Befehl ein. Das verursacht später schwerwiegende Anmeldeprobleme!

4

Prüfen ob Paperclip läuft

TERMINAL – STATUS PRÜFEN

```
docker logs paperclip --tail 5  
  
# Wenn alles klappt, siehst du: INFO: Server listening on 0.0.0.0:3100
```



FERTIG

Öffne jetzt `http://DEINE-IP:3100` im Browser. Du siehst die Paperclip-Anmeldeseite. Erstelle dort deinen Admin-Account.



Screenshot: Paperclip Login-Seite

Die Anmeldeseite nach der ersten Installation

SCHRITT 2 · DER WICHTIGSTE SCHRITT!

2 Claude Max anmelden

Das ist der trickreichste Teil – hier haben viele Leute Probleme. Wenn du die Anleitung genau befolgst, klappt es beim ersten Versuch.



WARUM IST DAS KOMPLEX?

Claude Code (das Programm im Container) speichert die Anmeldedaten als Datei auf dem Server. Diese Datei wird automatisch mit falschen Zugriffsrechten erstellt. Wir müssen das einmalig korrigieren.

1 Login-Befehl eingeben

Im Coolify-Terminal eingeben:

```
☰ TERMINAL
```

```
docker exec -it paperclip claude auth login
```

Es erscheint eine lange URL, die mit `https://claude.com/cai/oauth/...` beginnt.



Screenshot: Terminal zeigt OAuth-URL

Das Terminal wartet auf "Paste code here if prompted >"

2

URL öffnen und autorisieren

Kopiere die URL aus dem Terminal und öffne sie in deinem Browser. Melde dich mit deinem Claude-Account an und klicke auf "Authorize".



Screenshot: Claude OAuth-Autorisierungsseite

Die Anthropic-Seite fragt nach der Genehmigung für Claude Code

3

Code sofort ins Terminal einfügen

Nach der Autorisierung siehst du einen Code im Browser (in der URL-Leiste nach # oder auf der Seite). Dieser Code sieht ungefähr so aus:

BEISPIEL CODE-FORMAT

```
kSlH5k2DozJBQvG0...#m0n6eetnwpUZ9C6C...
```



60-SEKUNDEN-REGEL – ENTSCHEIDEND!

Der Code ist nur ca. 60 Sekunden gültig. Klicke direkt ins Coolify-Terminal-Fenster und füge den Code ein (Strg+V oder Cmd+V).

SCHICKE DEN CODE NICHT ÜBER CHAT, E-MAIL ODER ANDERE APPS

– bis der Code dort ankommt, ist er abgelaufen!

Nach dem Einfügen Enter drücken. Das Terminal zeigt:

ERFOLG

Login successful.

4

Zugriffsrechte korrigieren Pflicht!

Dieser Schritt ist immer notwendig. Die Anmelde-Datei wird als "root" gespeichert und muss für Paperclip lesbar gemacht werden:

TERMINAL – RECHTE KORRIGIEREN

```
# Volume-Pfad automatisch ermitteln VOL=$(docker inspect paperclip --  
format '{{range .Mounts}}{{.Source}}{{end}}')  
  
# Rechte setzen chmod 644 $VOL/.claude/.credentials.json chown -R  
1000:1000 $VOL/.claude/  
  
# Prüfen ob es geklappt hat docker exec paperclip claude auth status
```



SO SIEHT ERFOLG AUS

Der Status-Befehl gibt zurück: "loggedIn": true und
"subscriptionType": "max"



Screenshot: Terminal – "loggedIn: true"

Die JSON-Ausgabe von claude auth status zeigt subscriptionType: max

SCHRITT 3

3

Ersten Agenten erstellen

1

Agents → New Agent

In Paperclip links auf **Agents** klicken, dann oben rechts auf **+ New Agent**.



Screenshot: Agents-Seite mit "+ New Agent" Button

2

"Configure a runtime manually" wählen



Screenshot: "Add a new agent" Dialog mit drei Optionen

3

Claude Code (local) auswählen

Formular ausfüllen:

Feld	Wert
Agent name	Tester
Title	Test Agent
Adapter type	Claude Code (local)
Model	Claude Sonnet 4.5

Auf **Create agent** klicken.



Screenshot: New Agent Formular ausgefüllt

4

Agent-Instruktionen vereinfachen Empfohlen

Die Standard-Instruktionen des Agenten sind sehr lang und verlangsamen jeden Durchlauf erheblich. Wir ersetzen sie durch eine kurze, effiziente Version.

▼ Technische Schritte anzeigen

Im Terminal eingeben (UUID aus der Paperclip-URL kopieren):

```
TERMINAL – AGENTS.MD OPTIMIEREN
```

```
docker exec paperclip sh -c 'printf \ "You are a fast test
agent.\n\nDo NOT use bash.\nOnly use the paperclip
skill.\n\nSteps:\n1. Read the issue\n2. Answer immediately\n3.
Post answer as comment\n4. Mark issue as done\n" \ >
/paperclip/instances/default/companies/[COMPANY-
ID]/agents/[AGENT-UUID]/instructions/AGENTS.md'
```

Company-ID und Agent-UUID findest du in der Paperclip-URL



ERGEBNIS

Input-Tokens reduzieren sich von 10.000+ auf unter 50. Das macht jeden Run 3–4× schneller.

SCHRITT 4 · OPTIONAL

4

Codex-Agent (OpenAI) einrichten

Zusätzlich zu Claude kannst du auch OpenAI's Codex-Modell (GPT-5.3-Codex) nutzen – über dein ChatGPT-Abo, ohne API-Kosten.

1

In ChatGPT: Device Code aktivieren

Öffne ChatGPT → **Einstellungen** → und aktiviere "Autorisierung per Gerätecode für Codex".



SICHERHEITSHINWEIS VON OPENAI

Gerätecodes sind ein häufiges Phishing-Ziel. Gib den Code, den du gleich bekommst, niemals an andere weiter!

2

Codex-Agenten in Paperclip erstellen

Genau wie bei Claude, aber diesmal **Codex (local)** als Adapter und `gpt-5.3-codex` als Modell auswählen.

3

Codex anmelden

Im Terminal:

```
TERMINAL
```

```
docker exec -it paperclip codex login --device-auth
```

Es erscheint ein **URL + Code** (z.B. 0Y1W-QSIX1, 15 Minuten gültig).

1. URL öffnen: <https://auth.openai.com/codex/device>
2. Mit ChatGPT-Account anmelden
3. Code eingeben
4. Terminal zeigt: **Successfully logged in**

4

Zugriffsrechte für Codex korrigieren

```
TERMINAL
```

```
VOL=$(docker inspect paperclip --format '{{range .Mounts}}{{.Source}}
{{end}}') chown -R 1000:1000 $VOL/.codex/ chmod 644
$VOL/.codex/auth.json
```

⚡ HEARTBEAT DEAKTIVIEREN!

Bei Codex unbedingt den Heartbeat (automatische Intervall-Runs) deaktivieren oder den Agenten bei Nicht-Nutzung pausieren. Ohne diese Maßnahme explodieren die Token-Zahlen ins Hunderttausende!

SCHRITT 5 · LOSLEGEN!

5 Erste Aufgabe zuweisen

1 Agenten auswählen und "Assign Task" klicken



Screenshot: Tester-Dashboard mit "Assign Task" Button

2 Titel eingeben und "Create Issue"

Schreibe eine einfache Test-Aufgabe in den Titel, z.B. *"Was ist die Hauptstadt von Frankreich?"*



Screenshot: "New Issue" Dialog mit eingetragener Aufgabe

3

Live-Run beobachten

Im Dashboard siehst du direkt, dass der Agent aktiv ist (**running**). Unter **Runs** kannst du Schritt für Schritt verfolgen, was der Agent denkt und tut.



Screenshot: Runs-Seite mit "succeeded" und Token-Statistiken

Input: 47 tokens · Duration: 53s · Issue: done 

4

Ergebnis im Inbox prüfen

Alle Ergebnisse und Antworten des Agenten erscheinen in der **Inbox** (linkes Menü).



Screenshot: Inbox mit abgeschlossenem Task (grüner "done" Badge)

TROUBLESHOOTING

Bekannte Probleme & Lösungen

▼ ❌ "Login failed: Request failed with status code 400"

Ursache: Der OAuth-Code ist abgelaufen (Zeitlimit: ~60 Sekunden) oder wurde über einen Umweg übertragen.

Lösung: Befehl nochmal ausführen und den Code diesmal direkt und sofort ins Terminal einfügen – kein Chat, keine E-Mail dazwischen.



TIPP

Habe zwei Browser-Tabs offen: einen mit dem Coolify-Terminal, einen für die Claude-Autorisierungs-URL. So musst du nur zwischen Tabs wechseln.

▼ ❌ Agent meldet "Not logged in · Please run /login"

Ursache: Die Credentials-Datei hat falsche Berechtigungen – das passiert nach jedem Login.

Lösung:

```
☰ TERMINAL – IMMER NACH DEM LOGIN AUSFÜHREN!
```

```
VOL=$(docker inspect paperclip --format '{{range .Mounts}}{{.Source}}
{{end}}') chmod 644 $VOL/.claude/.credentials.json chown -R 1000:1000
$VOL/.claude/
```

▼ ❌ Agent meldet "blocked" – Bash permission error

Ursache: Claude Code versucht Terminal-Befehle (bash) auszuführen, die im Container nicht erlaubt sind.

Was ist "bash"? Ein Programm, das Terminal-Kommandos ausführt (z.B. Dateien lesen, Prozesse starten). Im Paperclip-Container ist das aus Sicherheitsgründen gesperrt.

Lösung: In der AGENTS.md explizit verbieten:

```
IN AGENTS.MD EINFÜGEN
```

```
Do NOT use bash or shell commands. NEVER. Only use the paperclip skill.
```

Oder: Berechtigungsproblem im `.claude`-Verzeichnis lösen:

```
📄 TERMINAL
```

```
VOL=$(docker inspect paperclip --format '{{range .Mounts}}{{.Source}}\n{{end}}') chown -R 1000:1000 $VOL/.claude/
```

▼ ❌ Codex "Permission denied (os error 13)"

Ursache: Codex-Credentials wurden als root-Nutzer gespeichert und sind für Paperclip nicht lesbar.

Lösung:

```
📄 TERMINAL
```

```
VOL=$(docker inspect paperclip --format '{{range .Mounts}}{{.Source}}\n{{end}}') chown -R 1000:1000 $VOL/.codex/ chmod 644 $VOL/.codex/auth.json
```

▼ ❌ Codex verbraucht hunderttausende Tokens

Ursache: Der Heartbeat-Mechanismus lädt bei jedem automatischen Check die gesamte bisherige Konversationsgeschichte. Diese wächst und wächst.

Lösung: Codex-Agenten immer pausieren, wenn er nicht aktiv genutzt wird:



Screenshot: "Pause" Button oben rechts in der Agenten-Ansicht

Alternativ: In der Agenten-Konfiguration (**Configuration**) den *Heartbeat on interval* deaktivieren.

BEST PRACTICES

Effizienz-Tipps

Konfiguration	Input Tokens	Dauer	Ergebnis
Standard AGENTS.md (ab Werk)	10.000+	2–5 Minuten	❌ often blocked
Minimale AGENTS.md	~50	~1 Minute	✅ done
Minimal + Bash-Verbot + chmod	~47	~53 Sekunden	✅ done

Die 5 goldenen Regeln

1. **AGENTS.md kurz halten** – jede zusätzliche Zeile kostet Token und Zeit
2. **"Do NOT use bash"** immer in die Instruktionen – verhindert das Blocked-Problem
3. **chmod nach jedem Login** – `chown -R 1000:1000 $VOL/.claude/`
4. **Codex pausieren** wenn nicht aktiv genutzt
5. **Heartbeat deaktivieren** bei Test-Agenten (spart Token im Hintergrund)

Monatliche Kosten auf einen Blick

Komponente	Kosten	Notiz
Server (Hetzner CX22)	~5€	Reicht für mehrere Agenten
Claude Max Abo	~100\$	Alle Claude-Agenten inklusive
ChatGPT Pro (für Codex)	~200\$ <i>optional</i>	Codex-Agent inklusive
Gesamt (nur Claude)	~105€	Empfehlung für den Start

CHEAT SHEET

Alle wichtigen Befehle

📄 BEFEHLS-REFERENZ

```
# — Container ————— docker logs paperclip --tail
20 # Logs anzeigen docker restart paperclip # Neustart docker ps | grep
paperclip # Status prüfen
# — Hilfreich: VOL-Variable setzen ————— VOL=$(docker inspect
paperclip --format '{{range .Mounts}}{{.Source}}{{end}}')
# — Claude Auth ————— docker exec -it paperclip
claude auth login # Login docker exec paperclip claude auth status # Status
prüfen chmod 644 $VOL/.claude/credentials.json # Rechte fix chown -R
1000:1000 $VOL/.claude/ # Eigentümer fix
# — Codex Auth ————— docker exec -it paperclip
codex login --device-auth chown -R 1000:1000 $VOL/.codex/ chmod 644
$VOL/.codex/auth.json
```

